IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 09/886,239	§
Filed: June 20, 2001	§
Inventor(s): Jeffrey D. Washington, Michael Santori and Robert C. Young	§
Title: Collector Node for a Graphical Program	§

Examiner:	Vu, Kieu D.
Group/Art Unit:	2173
Atty. Dkt. No:	5150-48900

APPEAL BRIEF

Dear Sir or Madam:

Further to the Notice of Appeal filed April 4, 2006 and the Notice of Non-Compliant Appeal Brief mailed October 13, 2006, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-7, 9-15, and 17-24 are pending in the application. Claims 20-24 are withdrawn from consideration. Claims 3, 4, 11, and 12 stand objected to as being dependent upon rejected base claims. Claims 1, 2, 5-7, 9, 10, 13-15, and 17-19 stand rejected and are the subject of this appeal. A copy of the claims, as incorporating entered amendments and as on appeal, is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been filed subsequent to the amendment of December 2, 2005. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The subject matter of the present claims relates generally to the field of graphical software programming. A graphical program is typically created by including various function nodes or icons in a block diagram and interconnecting them, e.g., by drawing lines or wires between them. The resulting interconnected nodes visually indicate functionality of the graphical program, e.g., visually indicate a function or process performed by the graphical program during its execution.

More particularly, independent claim 1 is directed to a method for creating a graphical program that performs a numerical function. The method comprises displaying a node in a graphical program in response to user input, where the node is operable to perform a first numerical function. For example, a block diagram of the graphical program may be displayed on a computer monitor, and the node may be displayed in the block diagram, e.g., in response to user input selecting the node for inclusion in the block diagram. (*See, e.g., Specification p. 6, lines 16-19; p. 49, lines 3-6; Figures 27, 30, and 32*).

The method further comprises configuring the node to receive data values, in response to user input. For example, in one embodiment the node may be configured to receive the data values by connecting an input terminal of the node to a data source in the graphical program (e.g., connecting the input terminal of the node to an output terminal of another node in the graphical program). (*See, e.g., Specification p. 6, lines 20-23; Figures 30 and 32*).

The method further comprises configuring the node with criteria information in response to user input, where the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node. For example, the criteria information may specify which subset of data values received by the node to apply the first numerical function to. As an example, in one embodiment the user may specify a "sliding block" collection mode for the node, e.g., to indicate that the first numerical function is to be performed on a moving window of data points. (*See, e.g., Specification p. 6, lines 26-29; p. 47, lines 20-22; p. 47, line 26 – p. 48, line 12; Figure 28*).

The method further comprises executing the graphical program. During execution of the graphical program, the node receives a plurality of data values. The node maintains state information regarding the received data values, e.g., to track the data values that have been received. (*See, e.g., Specification p. 6, lines 8-10; p. 7, lines 5-16; p. 50, lines 4-7*).

The method further comprises the node determining a first data collection on which to perform the first numerical function. The first data collection is determined based on the criteria information and the state information. In other words, the criteria information specifies which subset of data to apply the first numerical function to, and the state information specifies the actual data values that have been received. As noted above, the first data collection comprises a subset, but not all, of the plurality of data values received. The method further comprises the node performing the first numerical function on the first data collection. (*See, e.g., Specification p. 6, line 1 – p. 7, line 16; Figures 30-33*).

Independent claim 9 is directed to a memory medium that stores program instructions executable to perform the method of claim 1, which, as summarized above, is directed to creating a graphical program that performs a numerical function. The method includes displaying a node in a graphical program in response to user input, where the node is operable to perform a first numerical function. For example, a block diagram of the graphical program may be displayed on a computer monitor, and the node may be displayed in the block diagram, e.g., in response to user input selecting the node for inclusion in the block diagram. (*See, e.g., Specification p. 6, lines 16-19; p. 49, lines 3-6; Figures 27, 30, and 32*).

The method further includes configuring the node to receive data values, in response to user input. For example, in one embodiment the node may be configured to receive the data values by connecting an input terminal of the node to a data source in the graphical program (e.g., connecting the input terminal of the node to an output terminal of another node in the graphical program). (*See, e.g., Specification p. 6, lines 20-23; Figures 30 and 32*).

The method further includes configuring the node with criteria information in response to user input, where the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node. For example, the criteria information may specify which subset of data values received by the node to apply the first numerical function to. As an example, in one embodiment the user may specify a "sliding block" collection mode for the node, e.g., to indicate that the first numerical function is to be performed on a moving window of data points. (*See, e.g., Specification p. 6, lines 26-29; p. 47, lines 20-22; p. 47, line 26 – p. 48, line 12; Figure 28*).

The method further includes executing the graphical program. During execution of the graphical program, the node receives a plurality of data values. The node maintains state information regarding the received data values, e.g., to track the data values that have been received. (*See, e.g., Specification p. 6, lines 8-10; p. 7, lines 5-16; p. 50, lines 4-7*).

The method further includes the node determining a first data collection on which to perform the first numerical function. The first data collection is determined based on the criteria information and the state information. In other words, the criteria information specifies which subset of data to apply the first numerical function to, and the state information specifies the actual data values that have been received. As noted above, the first data collection comprises a subset, but not all, of the plurality of data values received. The method further includes the node performing the first numerical function on the first data collection. (*See, e.g., Specification p. 6, line 1 – p. 7, line 16; Figures 30-33*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 2, 5, 6, 9, 10, 13, 14, and 17 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Roach et al. (U.S. Patent No. 6,343,292, hereinafter "Roach") and Guttag et al. (U.S. Patent No. 4,933,878, hereinafter "Guttag").

Claims 7, 15, 18, and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Roach, Guttag, and Rogers et al. (U.S. Patent No. 5,497,500, hereinafter "Rogers").

VII. ARGUMENT

Section 103(a) Rejections Based on Roach and Guttag

Claims 1, 2, 5, 6, 9, 10, 13, 14, and 17 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Roach and Guttag. Appellant respectfully traverses these rejections for the following reasons. Different claims are addressed under respective subheadings.

Claims 1, 9

Appellant first respectfully reminds the Board that, "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." *In re Oetiker*, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). See also *In re Deminski*, 796 F.2d 436, 230 USPQ 313 (Fed. Cir. 1986); *In re Clay*, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992). However, the Guttag reference is not in the field of Appellant's endeavor and is not reasonably pertinent to the subject matter recited in the present claims.

The present claims relate to the field of graphical programming. As described in the Description of the Related Art section of the present application, a graphical program is created by placing a plurality of nodes or icons in a block diagram and interconnecting the nodes or icons, e.g., such that the interconnected plurality of nodes or icons visually indicates functionality of the graphical program. For example, claim 1 recites a method for creating a graphical program that performs a numerical function.

Guttag, on the other hand, teaches that, "The present invention relates to the field of computer graphics. In particular, this invention relates to the field of bit mapped computer graphics in which the computer memory stores data for each individual picture element or pixel of the display at memory locations that correspond to the location of that pixel on the display" (Col. 1, line 66 – Col. 2, line 3). The field of computer graphics is not at all the same as the field of graphical programming. <u>Guttag nowhere teaches or even remotely suggests the concept of a graphical program</u>. Thus, Appellant respectfully

submits that Guttag is not in the field of Appellant's endeavor and is not reasonably pertinent to the particular problem with which the inventors were concerned, and therefore is not available as a reference in a 103(a) rejection.

Furthermore, Appellant respectfully submits that the cited references do not teach the limitations recited in the present claims, as would be required to form a case of *prima facie* obviousness. For example, claim 1 recites in pertinent part, "configuring the node with criteria information in response to user input, wherein the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node". Roach and Guttag, taken either singly or in combination, do not teach this limitation of claim 1. The Examiner relies on Guttag to teach this limitation. However, as discussed above, Guttag does not even teach the concept of a graphical program or a node in a graphical program. Thus, Guttag certainly does not teach the concept of criteria information which indicates that a numerical function is to be performed on a subset, but not all, of the data values received by a node in a graphical program.

Appellant also respectfully disagrees with the Examiner's assertion that Roach teaches the node maintaining state information regarding received data values and using the state information to determine the data collection on which to perform the numerical function. The Examiner refers to the data structure for the SIB function that performs an addition of two parameters. The "numerical ranges" to which the Examiner refers do not constitute state information used to determine a data collection on which to perform the addition function. Roach teaches that the addition function is performed on the two parameters, and the numerical ranges are simply used to validate that the values of the two parameters fall within a certain range of values (See Col. 5, lines 25-41 and FIG. 4).

Furthermore, claim 1 recites, "the node determining a first data collection on which to perform the first numerical function based on the criteria information and the state information." In other words, in claim 1, the first data collection is determined based on both the state information and the criteria information. As discussed above, the cited references do not teach configuring the node with criteria information which indicates that the numerical function is to be performed on a subset, but not all, of the data values received by the node. The cited references also do not teach, "the node

determining a first data collection on which to perform the first numerical function based on the criteria information and the state information." Appellant also notes that the Examiner's explanation of the rejection of claim 1 does not take into account the limitation that the criteria information is used in determining the first data collection.

Thus, for at least the reasons set forth above, Appellant respectfully submits that claim 1 is patentably distinct over the cited references. Inasmuch as independent claim 9 recites similar limitations as those discussed above with respect to claim 1, Appellant respectfully submits that claim 9 is also patentably distinct over the cited references.

Claims 2 and 10

Claims 2 and 10 recite additional limitations not taught by the cited references. For example, claim 2 recites the additional limitations:

> receiving user input requesting to specify configuration information for the node;
> displaying a graphical user interface (GUI) for specifying configuration information for the node, in response to the user input requesting to specify configuration information for the node;

The Examiner asserts that Roach teaches these limitations, citing Col. 3, lines 59-67. Appellant respectfully disagrees. This portion of Roach reads as follows:

> Specifically, upon receiving a request for creating a new service for a subscriber or modifying an existing subscriber service, a user, by employing GUI 4, is presented a template on which a graphical representation of the subscriber service, such as a logic diagram, may be described. The service development tool may provide to the user a library of existing SIB primitives, each of which performs a single, unique function. The user may select and place one or more SIB icons on the template, modify parameters pertaining to the icons, and interrelate the icons by creating line segments therebetween. By creating a service logic diagram in this way, the user is capable of fully describing the operation of the desired subscriber service without writing a single line of program code. An illustration of a logic diagram for an exemplary subscriber service is shown in FIG. 2.

Thus, Roach is here describing a graphical user interface for use in creating a logic diagram, e.g., such as shown in FIG. 2. However, claim 2 recites the limitations of receiving user input requesting to specify configuration information <u>for a particular node in the graphical program</u> and displaying a GUI for specifying configuration information

for the node in response to this request. Roach teaches nothing at all in the cited passage about receiving user input requesting to specify configuration information <u>for a particular node</u> (or for a particular SIB icon in the logic diagram) and then displaying a GUI in response to the request.

Furthermore, claim 2 also recites the limitation of, "wherein said configuring the node with criteria information in response to user input comprises configuring the node with the criteria information in response to user input received via the GUI." The Examiner asserts that, "Roach further teach configuring the node with the criteria information (addition, subtraction, multiplication, or division)(col 5, lines 25-40)." However, simply configuring a node to perform addition, subtraction, multiplication, or division is not at all the same with configuring the node with criteria information such as recited in claim 1. (Claim 1 recites that, "the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node".)

Appellant also notes that the Examiner admits with respect to claim 1 that, "Roach does not teach that the numerical function is to be performed on a subset, but not all of the received data values." Thus, it is difficult to ascertain why in the rejection of claim 2 the Examiner asserts that Roach teaches configuring the node with criteria information which indicates that the numerical function is to be performed on a subset, but not all, of the data values received by the node.

Appellant thus submits that the cited references, taken either singly or in combination, do not teach the limitations recited in claims 2 and 10.

<u>Claim 17</u>

Claim 17 recites additional limitations not taught by the cited references:

> wherein the criteria information specifies a number of data values on which to perform the first numerical function;
> wherein said node receiving the plurality of data values during execution of the graphical program comprises the node receiving a greater number of data values than the number of data values specified by the criteria information;
> wherein the number of data values in the first data collection is equal to the number of data values specified by the criteria information.

As discussed above with respect to claim 1, Guttag does not teach the concepts of a graphical program, or a node in a graphical program, or criteria information which indicates that a numerical function is to be performed on a subset, but not all, of the data values received by a node in a graphical program. Thus, Guttag also does not teach these further limitations.

Section 103(a) Rejections Based on Roach, Guttag, and Rogers

Claims 7, 15, 18, and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Roach, Guttag, and Rogers. Appellant respectfully traverses these rejections for the following reasons. Different claims are addressed under respective subheadings.

Claims 7 and 15

Claims 7 and 15 recite the following additional limitations:

> wherein said configuring the node to receive data values comprises connecting a wire to an input terminal of the node from an output terminal of another node in the graphical program, in response to user input;
>
> wherein said node receiving the plurality of data values during execution of the graphical program comprises the node receiving the plurality of data values via the wire connected to the input terminal of the node.

The Examiner relies on Rogers to teach these additional limitations. As the Board is certainly aware, "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success <u>must both be found in the prior art</u>, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)" as stated in the MPEP §2142 (*emphasis added*).

As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the

claimed invention is defective as hindsight analysis. Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

The Examiner asserts that, "it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Rogers' teaching in Roach's system with the motivation being to receive data through the connected wire." However, this argument is similar to saying that, "Rogers teaches feature A. Therefore, it would be obvious to combine Rogers with Roach in order for Roach to achieve feature A." This is clearly simply hindsight analysis using Appellant's claim as a blueprint. The Examiner has not shown that the prior art contains a clear and particular teaching or suggestion for combining Rogers and Roach. Thus, Appellant respectfully submits that a case of *prima facie* obviousness has not been established with respect to claims 7 and 15.

Claim 18

Claim 18 recites additional limitations not taught by the cited references, as follows:

> wherein the node is further operable to perform a second numerical function;
>
> wherein the method further comprises the node performing the second numerical function on the first data collection, in addition to performing the first numerical function on the first data collection.

The Examiner asserts that these limitations are taught by Rogers, stating that, "Rogers teaches performing different operations (addition, subtraction) on the same data collection (Fig. 75)." However, in FIG. 75, the addition operation is performed by one node (the "ADD" node), and the subtraction operation is performed by another node (the "SUBTRACT" node). In other words, the two numerical functions are performed by two

different nodes. In contrast, claim 18 recites that two numerical functions are both performed by the same node.

Appellant thus submits that the cited references, taken either singly or in combination, do not teach the limitations recited in claim 18.

Claim 19

Claim 19 recites additional limitations not taught by the cited references, as follows:

> 19. (Previously Presented) The method of claim 18,
>
> wherein the node includes a first output terminal for outputting a result of performing the first numerical function on the first data collection and a second output terminal for outputting a result of performing the second numerical function on the first data collection.

In the rejection of claim 19, the Examiner again refers to FIG. 75 of Rogers. However, as discussed above with respect to claim 18, Rogers teaches that the two numerical functions are performed by two different nodes. Rogers does not teach that a single node includes two different output terminals, one for outputting a result of a first numerical function, and another for outputting a result of a second numerical function.

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1, 2, 5-7, 9, 10, 13-15, and 17-19 was erroneous, and reversal of the Examiner's decision is respectfully requested.

The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-48900/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: October 24, 2006 JCH/JLB

VIII. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments and as on appeal.

1. (Previously Presented) A computer-implemented method for creating a graphical program that performs a numerical function, the method comprising:

displaying a node in a graphical program in response to user input, wherein the node is operable to perform a first numerical function;

configuring the node to receive data values, in response to user input;

configuring the node with criteria information in response to user input, wherein the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node;

executing the graphical program;

the node receiving a plurality of data values during execution of the graphical program, wherein the node maintains state information regarding the received data values;

the node determining a first data collection on which to perform the first numerical function based on the criteria information and the state information, wherein the first data collection comprises a subset, but not all, of the plurality of data values received; and

the node performing the first numerical function on the first data collection.

2. (Previously Presented) The method of claim 1, further comprising:

receiving user input requesting to specify configuration information for the node;

displaying a graphical user interface (GUI) for specifying configuration information for the node, in response to the user input requesting to specify configuration information for the node;

wherein said configuring the node with criteria information in response to user input comprises configuring the node with the criteria information in response to user input received via the GUI.

3. (Previously Presented) The method of claim 1,

wherein the node is configurable with a plurality of collection modes, wherein each collection mode defines a subset of received data values on which to perform the first numerical function;

wherein said configuring the node with the criteria information in response to user input comprises configuring the node with a first collection mode in response to user input selecting the first collection mode from the plurality of collection modes;

wherein said node determining the first data collection based on the criteria information and the state information comprises the node determining the first data collection based on the first collection mode and the state information.

4. (Previously Presented) The method of claim 3,

wherein the node is configurable with one or more of the following collection modes:

Sliding Block; and

Fixed Block.

5. (Original) The method of claim 1,

wherein the node is a primitive node provided by a graphical programming development environment for inclusion in the graphical program.

6. (Previously Presented) The method of claim 1,

wherein the first numerical function performed on the first data collection comprises one of:

a numerical average function;

a summation function;

a minimum value function;

a maximum value function.

7. (Previously Presented) The method of claim 1,

wherein said configuring the node to receive data values comprises connecting a wire to an input terminal of the node from an output terminal of another node in the graphical program, in response to user input;

wherein said node receiving the plurality of data values during execution of the graphical program comprises the node receiving the plurality of data values via the wire connected to the input terminal of the node.

8. (Canceled)

9. (Previously Presented) A memory medium for creating a graphical program that performs a numerical function, the memory medium comprising program instructions executable to:

display a node in a graphical program in response to user input, wherein the node is operable to perform a first numerical function;

configure the node to receive data values, in response to user input;

configure the node with criteria information in response to user input, wherein the criteria information indicates that the first numerical function is to be performed on a subset, but not all, of the data values received by the node;

wherein during execution of the graphical program, the node is operable to:

receive a plurality of data values;

maintain state information regarding the received data values;

determine a first data collection on which to perform the first numerical function based on the criteria information and the state information, wherein the first data collection comprises a subset, but not all, of the plurality of data values received; and

perform the first numerical function on the first data collection.

10. (Previously Presented) The memory medium of claim 9, further comprising program instructions executable to:

receive user input requesting to specify configuration information for the node;

display a graphical user interface (GUI) for specifying configuration information for the node, in response to the user input requesting to specify configuration information for the node;

wherein said configuring the node with criteria information in response to user input comprises configuring the node with the criteria information in response to user input received via the GUI.

11. (Previously Presented) The memory medium of claim 9,

wherein the node is configurable with a plurality of collection modes, wherein each collection mode defines a subset of received data values on which to perform the first numerical function;

wherein said configuring the node with the criteria information in response to user input comprises configuring the node with a first collection mode in response to user input selecting the first collection mode from the plurality of collection modes;

wherein said node determining the first data collection based on the criteria information and the state information comprises the node determining the first data collection based on the first collection mode and the state information.

12. (Previously Presented) The memory medium of claim 11,

wherein the node is configurable with one or more of the following collection modes:

Sliding Block; and

Fixed Block.

13. (Original) The memory medium of claim 9,

wherein the node is a primitive node provided by a graphical programming development environment for inclusion in the graphical program.

14. (Previously Presented) The memory medium of claim 9,

wherein the first numerical function performed on the first data collection comprises one of:

a numerical average function;

a summation function;

a minimum value function;

a maximum value function.

15. (Previously Presented) The memory medium of claim 9,

wherein said configuring the node to receive data values comprises connecting a wire to an input terminal of the node from an output terminal of another node in the graphical program, in response to user input;

wherein said node receiving the plurality of data values during execution of the graphical program comprises the node receiving the plurality of data values via the wire connected to the input terminal of the node.

16. (Canceled)

17. (Previously Presented) The method of claim 1,

wherein the criteria information specifies a number of data values on which to perform the first numerical function;

wherein said node receiving the plurality of data values during execution of the graphical program comprises the node receiving a greater number of data values than the number of data values specified by the criteria information;

wherein the number of data values in the first data collection is equal to the number of data values specified by the criteria information.

18. (Previously Presented) The method of claim 1,

wherein the node is further operable to perform a second numerical function;

wherein the method further comprises the node performing the second numerical function on the first data collection, in addition to performing the first numerical function on the first data collection.

19. (Previously Presented) The method of claim 18,

wherein the node includes a first output terminal for outputting a result of performing the first numerical function on the first data collection and a second output terminal for outputting a result of performing the second numerical function on the first data collection.

20. (Withdrawn) A memory medium for creating a graphical program that performs a numerical function, the memory medium comprising program instructions executable to:

display a node in a graphical program in response to user input, wherein the node is operable to perform a first numerical function;

configure the node to receive data values, in response to user input; and

configure the node with information specifying a number N of data values on which to perform the first numeric function, in response to user input specifying the number N;

wherein during execution of the graphical program, the node receives a plurality of data values;

wherein each time the node receives a data value during execution of the graphical program, the node is operable to:

determine whether the node has received at least N data values; and

perform the first numerical function on the N most recently received data values if the node has received at least N data values.

21. (Withdrawn) The memory medium of claim 20,

wherein the node is a primitive node provided by a graphical programming development environment for inclusion in the graphical program.

22. (Withdrawn) A memory medium for creating a graphical program that performs a numerical function, the memory medium comprising program instructions executable to:

display a node in a graphical program in response to user input, wherein the node is operable to perform a first numerical function;

connect a wire to the node in response to user input, wherein connecting the wire to the node configures the node to receive data values via the wire; and

configure the node with information specifying a number N of data values on which to perform the first numeric function, in response to user input specifying the number N;

wherein during execution of the graphical program, the node receives a plurality of data values via the wire;

wherein the node is operable to:

determine when a first set of N data values have been received via the wire; and

perform the first numerical function on the first set of N data values.

23. (Withdrawn) The memory medium of claim 22,

wherein the node is further operable to:

determine when a second set of N data values have been received, wherein the second set of N data values comprises N data values received after the first set of N data values; and

perform the first numerical function on the second set of N data values.

24. (Withdrawn) The memory medium of claim 22,

wherein the node is a primitive node provided by a graphical programming development environment for inclusion in the graphical program.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.